IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 7, 2019, accepted February 19, 2019, date of current version May 10, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2904117*

# 3D Motion Reconstruction From 2D Motion Data Using Multimodal Conditional Deep Belief Network

**MUHAMMAD JAVAD HEYDARI[1] AND SAEED SHIRY GHIDARY[2]**
[1]Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran
[2]Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran

Corresponding author: Saeed Shiry Ghidary (shiryghidary@gmail.com)

**ABSTRACT** In this paper, we propose a deep generative model named multimodal conditional deep belief network (MCDBN) for cross-modal learning of 3D motion data and their non-injective 2D projections on the image plane. This model has a three-sectional structure, which learns conditional probability distribution of 3D motion data given 2D projections. Two distinct conditional deep belief networks (CDBNs), encode the real-valued spatiotemporal patterns of 2D and 3D motion time series captured from the subjects' movements into compact representations. The third part includes a multimodal restricted Boltzmann machine which in the training process, learns the relationship between the compact representations of data modalities by variation information criteria. As a result, conditioned on a 2D motion data obtained from a video, MCDBN can regenerate 3D motion data in the generation phase. We introduce Pearson correlation coefficient of the ground truth and the regenerated the motion signals as a new evaluation metric in motion reconstruction problems. The model is trained with human motion capture data and the results show that the real and the regenerated signals are highly correlated, which means the model can reproduce the dynamical patterns of the motion accurately.

**INDEX TERMS** Motion analysis, signal reconstruction, artificial neural networks, time series analysis.

## I. INTRODUCTION

Regenerating 3D human motion from 2D projections of body landmarks on the image plane is a challenging task with multiple applications such as interactive human-robot interfaces, computer graphics, and virtual reality. 3D motion data contain frame-wise 3D positions of the human joints in the real world which are usually referred to as 3D pose. Similarly, the sequences of the 2D pixel-wise address of body landmarks on the image plane or 2D poses form the 2D motion data. The depth values of 3D poses (distances of joints from the camera) are lost when they are projected onto the image plane. Therefore, 3D motion reconstruction from 2D motion data is an ill-posed inverse problem; meaning a single 2D pose may concur with infinite 3D poses as a solution of the non-injective projection. Of course, not all of these solutions are probable or physically feasible. So, considering the extra information and imposing physical constraints can limit the space of solutions. Any independent channel of sensory input/output between a human and the world is referred to as modality. Humans can learn to regenerate any arbitrary trajectory from demonstration via cross-modal processing. If a person regenerates a special trajectory many times, the produced trajectories are not exactly the same, but they are very similar. It can be supposed that humans learn a multimodal generative model and regenerate a new trajectory through sampling from this model.

Most of the previous methods do not consider the whole 2D and 3D motion time series and only try to find a model that can estimate a single 3D pose from corresponding 2D pose [1]. Likewise, utilizing tools such as deterministic neural networks [2], mathematical optimizers [3] and parameter estimators [4] lead to the deterministic models. It means these models reproduce exactly the same output, providing a fixed input. However, in many other applications such as virtual reality, robotics, and animation, models with the capability of regenerating accurate movements with a controlled level of stochasticity are desired. Human movements are rich spatiotemporal data with many variations and this stochasticity

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

causes the regenerated motions to seem more humanlike and realistic.

Providing the ability to model the high dimensional time series of 2D and 3D motion data with complex dependencies requires a powerful machine learning tool. Generative models are among the best candidates. They are able to learn the spatiotemporal patterns of motion data and regenerate new variations of motions while preserving a certain style via their sampling procedure. The recent successes of deep generative models in handling multimodal temporal data with complicated non-linear interactions persuade us to focus on deep generative models in this study.

In this paper, a multimodal deep generative architecture is proposed to capture the cross-modal relation of 3D and 2D motion data modalities. We refer to the proposed model as Multimodal Conditional Deep Belief Network (MCDBN). MCDBN learns the cross-modal relation of data modalities in the training phase. As a result, it would be able to regenerate 3D motions conditioned on 2D motion data during the generation process. The model can be used in the vast area of applications including imitation learning and human character generation.

MCDBN is inspired from the mentioned ability of the humans in recreating 3D motions that are seen in a video upon prior knowledge and intuition about kinematic constraints [5]. Most of the previous approaches learn priors explicitly through the ground truth data [6]. As mentioned, there is an infinite 3D pose whose projections match a single 2D pose. But the motion data are highly structured since the body joints hinge on each other in a single frame, and also their trajectories are dependent across the frames. So, the solution space can be limited. MCDBN takes into account the prior knowledge and constraints implicitly via its weights and parameters by exploiting 2D -3D motion data coincidence patterns in the training data.

MCDBN acts as an estimator of the joint distribution of 2D and 3D motion data time series. It is trained as an encoder-decoder model that learns rich statistical relationships between 2D and 3D motion data in the form of a shared latent representation. Given 2D motion data as input, the 3D motion data are regenerated by sampling from the learned joint distribution. The key property of our approach is that it considers both anthropometric and kinematic constraints, simultaneously. In more details, the training process tunes the parameters of the model in such a way that MCDBN assigns a low probability to improbable and implausible poses.

We use the variation of information (VI) as the training criterion as an alternative to the common Negative Log-Likelihood (NLL). VI is a metric of informativeness of two random variables about each other [7], [8]. It is based on a simple linear expression of the mutual information. Using VI as a training criterion causes to learn more robust shared representation which maximizes the intra-modality association.

Our main contributions in this paper are:

- Propounding a new approach for 3D motion reconstruction from 2D motion data as a probabilistic cross-modal learning problem
- Proposing a new deep architecture for modeling the coupled time series
- Using variation information criteria instead of NLL for cross-modal learning of coupled time series
- Evaluating the presented model on realistic datasets and proposing new error generation measurement

Remaining of the paper is organized as follows. Some of the related papers are discussed in the next section. The required backgrounds are presented in Section III. Section IV provides the details of the proposed method and the training and the inference procedures. Our experimental results are presented in Section V and the last section includes a conclusion.

## II. RELATED WORK

3D pose estimation from 2D pose has been studied extensively in computer vision [9], [10], multimodal learning [11] and robotics [12]. In this section, some of the most relevant papers in robotics and multimodal deep learning literature are investigated.

Ngiam *et al.* [13] proposed an Autoencoder based [14] model to learn the multimodal features. The model has a three-step learning process. The main application of this paper was discriminating letters in audiovisual data. The proposed model outperforms unimodal structures only in noisy data. They believed unimodal clean data have less ambiguity and multimodal data help distinguishing letters in noisy situations.

Srivastava and Salakhutdinov [15] proposed a structure to reconstruct the missing modality in text and image namely Multimodal Deep Boltzmann Machine (MDBM). MDBM learns a joint distribution over the multimodal space and in the absence of one of the modalities, it estimates the other one. MDBM contains a modality-specific Deep Boltzmann Machine for each modality. One of the DBMs takes the binary coded text input and another one is a Gaussian RBM that is assigned to the image modality. A similar architecture called Multimodal Deep Belief Network (MDBN) has been proposed by Srivastava and Salakhutdinov [16], but the results have shown that MDBM outperforms MDBN. It seems the information flow manner causes this superiority. In MDBM, information streams both bottom-up and top-down, while MDBN has a one-sided flow of information. So, the task of multimodal modeling in MDBM is distributed in entire of the network while in MDBN, solely the top layer is responsible for learning the data association.

An Autoencoder based [14] computational framework for integration of sensory-motor time-series has been proposed by Noda *et al.* [11]. They implemented the framework in a humanoid robot for modalities including raw RGB images, sound spectrums, and joint angles. The framework has been examined for the ability of cross-modal memory retrieval

and time series prediction considering the root mean square (RMS) of the estimated signals and the ground truth values as an error measurement.

Aiming gesture detection and localization Neverova *et al.* [17] proposed a multiscale and multimodal deep architecture. Similar to the other deep models, careful initialization was one of their keys to success. They fused data modalities gradually through a technique that was named ModDrop [18]. Authors claimed that fusing multiple modalities at several spatial and temporal scales causes to a significant increase in the recognition rate, and provides the ability to compensate errors of the individual classifiers as well as noise in the separate channels. Likewise, it ensures the robustness of the classifier to missing signals in one or several channels to produce meaningful predictions from any number of available modalities.

According to Sohn *et al.* [19], having the ability of reasoning about missing data based on available data modalities is a mandatory condition for any multimodal model. So, they introduced a new representation learning framework that explicitly aims at this goal. They proposed to train the model by minimizing the variation of information rather than minimizing NLL. The paper is one of the main references of our work and we will discuss the novel learning criteria in the next sections.

RNN-RBM is a probabilistic model based on a recurrent neural network that has been proposed by Boulanger-Lewandowski *et al.* [20]. RNN-RBM has been applied in the music transcription problem [21]. It learns the relationships between the input and the output variables by several CRBMs whose parameters are tuned by a recurrent neural network. Villegas *et al.* [22] used a recurrent neural network for the motion regeneration. The proposed network consists of two main parts. The first part solves the forward kinematic problem, and the second part that is named cycle consistency learns to solve the inverse kinematic problem in an unsupervised manner.

For the multimodal gesture segmentation and recognition, Wu *et al.* [23] presented a semi-supervised hierarchical dynamic framework that is called Deep Dynamic Neural Networks (DDNN). DDNN is based on the HMM whose emission probability is obtained through a feedforward neural network. The input modalities of DDNN are skeletal information, depth, and RGB images. The model contains a Gaussian-Bernoulli Deep Belief Network for handling the skeletal dynamics and a 3D Convolutional Neural Network (3DCNN) for processing and fusing batches of depth and RGB.

Peng *et al.* [24] utilized deep Reinforcement Learning (RL) for regenerating the actions including locomotion, acrobatics, and martial arts from the video. Firstly, they used deep methods for the pose estimation. Then, the deep RL framework has been trained based on the estimated motion capture. They claimed that their method can predict the potential human motions from the still images. Lin and Amer [25] proposed a generative model for human motion modeling using Generative Adversarial Networks (GANs).

Bo and Sminchisescu [26] proposed twin Gaussian processes (TGP) for the structured prediction. TGP uses Gaussian process (GP) priors on both covariates and responses and estimates outputs by minimizing the Kullback-Leibler divergence between two GPs. They applied their proposed model on the 3D motion reconstruction problem. Many methods have been proposed for the motion modeling based on GP [27]. The main drawback of these approaches is the high computational complexity.

Amer *et al.* [28] proposed a temporal hybrid model (generative and discriminative) for classifying the sequential data from multiple heterogeneous modalities. According to the authors, such a hybrid model can exploit the power of discriminative classifiers along with the representation power of the generative models. With the aim of combining temporal discriminative and generative architectures into a unified single model, they have added a discriminative component to the CRBM and named the new architecture discriminative CRBM (DCRBM). A modality specific DCRBM has been trained for each data modality followed by training a fusion layer. The new structure has been named Multimodal Discriminative CRBMs (MMDCRBMs). To exploit the generative capability of MMDCRBMs, the model has been trained in such a way that it generates the lower-level data corresponding to the specific label that closely matches the actual input data. In an earlier version of this paper [29], the authors had used a Conditional Random Field (CRF) as a discriminative part in which the input representation to it had been extracted through a Conditional Deep Belief Network (CDBN).

## III. PRELIMINARIES

In this section, the required backgrounds are briefly summarized. We refer interested readers to some review articles [30] and references therein for more information.

### A. RESTRICTED BOLTZMANN MACHINE (RBM)

A Boltzmann Machine (BM) which is depicted in Fig. 1 can be viewed as either a probabilistic neural network or an undirected graphical model. It consists of visible and hidden

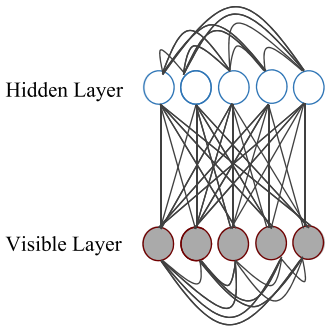


**FIGURE 1.** Boltzmann machine (BM).

layers of the binary variables with fully connected links [31], [32]. The variables in the visible layer represent the data while the hidden variables have the role of enlarging class of representable distributions. BM updates the values of variables according to a Bernoulli trial. As a result, in the limit of infinite time, the joint distribution of the visible and the hidden variables is the Gibbs-Boltzmann probability distribution.

The main extension of BM has been named Restricted Boltzmann Machine (RBM), owing to restricting the network connections to the inter-layer links and omitting the intra-layer ones. As it is depicted in Fig. 2, the RBM has a bipartite architecture in which the variables are connected through the symmetric undirected links that are fully connected between the layers. RBM learns a generative model of data distribution through an unsupervised method. Employing stochastic variables makes RBM less vulnerable to local minima and provides excellent generalization capability [30].

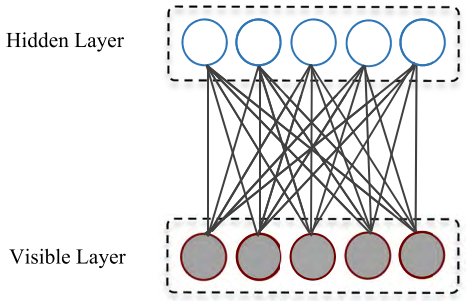


**FIGURE 2.** Restricted boltzmann machine (RBM).

Joint distribution of the visible and the hidden variables of a typical RBM is defined through an energy function which associates with a scalar energy to every possible configuration of the variables. Let $v = (v_1.v_2.\cdots.v_n)$ be the set of observed variables and $h = (h_1.h_2.\cdots.h_m)$ be the set of hidden variables, then an energy-based probabilistic model defines the joint distribution of visible and hidden variables as (1):

$$p(v,h) = \frac{1}{Z}\exp(-E(v,h)) \tag{1}$$

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{ij} v_i w_{ij} h_j \tag{2}$$

$$Z = \sum_{v,h} \exp(-E(v,h)), \tag{3}$$

where $i$ and $j$ are used to index the visible and the hidden variables respectively; $a_i$ and $b_j$ are the biases. $w_{ij}$ is the weight between the $i_{\text{th}}$ visible and the $j_{\text{th}}$ hidden variables and $Z$ is the partition function that normalizes the probabilities with respect to all possible configurations. The marginal and the conditional distributions of RBM are defined according to (4) to (7):

$$p(v) = \sum_h p(v,h) \tag{4}$$

$$p(h) = \sum_v p(v,h) \tag{5}$$

$$p(v\,|\,h) = \prod_i p(v_i\,|\,h) = \prod_i \sigma(a_i + \sum_j w_{ij} h_j) \tag{6}$$

$$p(h\,|\,v) = \prod_j p(h_j\,|\,v) = \prod_j \sigma(b_j + \sum_i w_{ij} v_i), \tag{7}$$

where $\sigma$ is the sigmoid function:

$$\sigma(x) = \left(\frac{1}{1+\exp?(-x)}\right). \tag{8}$$

To provide the ability of modeling real-valued data, the binary visible variables can be replaced by real-valued variables that have Gaussian distributions. So the energy of a Gaussian RBM (GRBM) is defined as (9):

$$E(v,h) = -\sum_i \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_j b_j h_j - \sum_{ij} \frac{v_i}{\sigma_i} w_{ij} h_j. \tag{9}$$

The conditional distributions of the visible and the hidden variables are as follows:

$$p(v_i\,|\,h) = \mathcal{N}\left(a_i + \sigma_i \sum_j w_{ij} h_j, \sigma_i^2\right) \tag{10}$$

$$p(h_i{=}\,1\,|\,v) = \sigma\left(b_i + \sum_i \frac{v_i}{\sigma_i} w_{ij}\right). \tag{11}$$

The parameters of RBM are trained by maximizing the likelihood function. But, maximum likelihood estimation is computationally intractable. The problem has been circumvented by applying Stochastic Gradient Descent (SGD) while approximating the gradient with Contrastive Divergence (CD) [33] or any variant of it like the Persistent CD (PCD) [34] CD follows the gradient via (12).

$$CD_n \propto D_{KL}(p_0(x)||p_\infty(x)) - D_{KL}(p_n(x)||(p_\infty)). \tag{12}$$

where $p_n$ is the distribution of a Markov chain running for $n$ steps and $D_{KL}$ symbolizes the Kullback-Leibler divergence. Bi-linearity of the energy function and lack of the inter-layer connections cause conditional independence of the hidden variables given the visible ones and vice versa. As a result, the computation of log-likelihood is more efficient and less expensive in RBM compared with BM.

The idea of stacking RBMs on top of each other makes two well-known deep generative models Deep Boltzmann Machine (DBM) [35] and Deep Belief Network (DBN) [36], [37]. DBM and DBN have similar architectures except that the DBM connections are undirected while DBN connections are directed except the top layer. It seems that having undirected connections makes two-way (bottom-up and top-down) inference in DBM much easier which is a favorable property when facing imperfect and missing data. Hinton *et al.* [38] proposed a greedy and layer-wise training procedure for DBN in which the hidden layers of lower RBMs are considered as visible data for upper RBMs.

### B. CONDITIONAL RESTRICTED BOLTZMANN MACHINE (CRBM)

The RBM only can model the static data. So far, several variants of the RBM like Temporal RBM [39], [40] and Conditional RBM (CRBM) [41] have been proposed for incorporating temporal information. CRBM is a biologically-inspired nonlinear generative model proposed for modeling the high dimensional time series [41]–[45]. It contains binary hidden and real-valued visible variables. In order to consider the temporal dynamics of data, CRBM incorporates long-term temporal dependencies by adding two types of directed autoregressive connections (Fig. 3). These two types of links connect the visible variables in the last $N$ frames to the current hidden variables and the visible variables in the last $M$ frames to the current visible variables , respectively. Taylor assumed $M = N$ and named it as the order of model. For ease of visualization, we show the fully connected links between layers with bold arrows. So, each arrow between two layers represents a set of links that connect all the variables of two layers together.

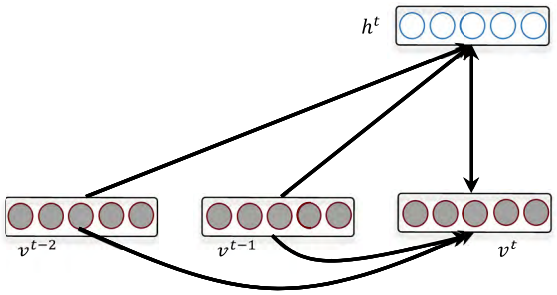


**FIGURE 3.** Conditional restricted boltzmann machine (CRBM).

The autoregressive connections are treated as dynamically changing biases, so the conditional independence assumptions of the RBM still hold, i.e. the conditional distributions of visible or hidden variables given the other variables factorizes completely. The conditional distribution of current real-valued visible variables $v^t$ and the current binary hidden variables $h^t$ given the history of visible variables from time step $t-1$ to $t-q$ ($H^t$), is obtained through the energy function as follows:

$$p\left(v^t, h^t \mid H^t\right) = \frac{1}{Z}\exp\left(-E\left(v^t, h^t \mid H^t\right)\right) \tag{13}$$

$$E\left(v^t, h^t \mid H^t\right) = \sum_i \frac{(v_i - \widehat{a}_i)^2}{2\sigma_i^2} - \sum_j \widehat{b}_j h_j - \sum_{ij} \frac{v_i}{\sigma_i} w_{ij} h_j \tag{14}$$

$$H^t = [v^{t-n}, v^{t-n+1}, \ldots, v^{t-1}] \tag{15}$$

$$\widehat{a}_i = a_i + \sum_k \sum_{q=1}^{n} A_{ki}^{t-q} v_k^{t-q} \tag{16}$$

$$\widehat{b}_j = b_j + \sum_k \sum_{q=1}^{m} B_{kj}^{t-q} v_k^{t-q}, \tag{17}$$

where $\widehat{a}_i$ and $\widehat{b}_j$ are the dynamic biases of the $i_{\text{th}}$ visible variable and the $j_{\text{th}}$ hidden variable, respectively; $v_k^{t-q}$ is the $k_{\text{th}}$ visible variable at time $t-q$ and $A_{ki}^{t-q}$ and $A_{kj}^{t-q}$ are the weights of connection from the $k_{\text{th}}$ visible variable at time $t-q$ to the $i_{\text{th}}$ current visible and the $j_{\text{th}}$ current hidden variables.

Similar to RBM, CRBMs can also be stacked on each other to create a Conditional Deep Belief Network (CDBN). A two-layer CDBN is depicted in Fig. 4. In this figure, the hidden layer of lower CRBM and the hidden layer of upper CRBM are denoted by $h1$ and $h2$, respectively. Besides, the superscripts are used to denote the time steps and fully connected links between layers are shown with bold arrows, for ease of visualization.

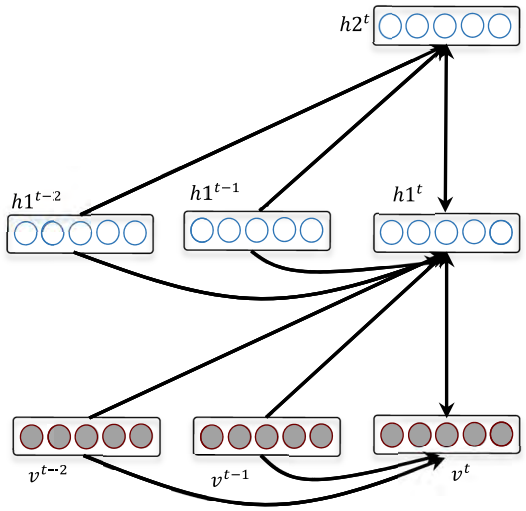


**FIGURE 4.** Conditional deep belief network (CDBN).

CDBN has a greedy layer-wise training algorithm. The partition function of CRBM ($Z$) is computationally intractable and the model parameters are learned through any variant of the CD. Top-down weights are used to regenerate the lower visible data during the inference. The visible data in each individual layer are produced by alternating Gibbs sampling between hidden ($h^t$) and visible states ($v^t$). To start alternating Gibbs sampling, it is necessary to initialize either $h^t$ or $v^t$. Taylor [41] decided to initialize $h^t$. They simply clamp the visible variables in the early steps and alternate between stochastically updating the hidden and the visible variables. Likewise, as explained, there are no connections between the variables of the same layer and the inference procedures are done in a parallel manner for all the visible and the hidden variables. The reader is referred to Taylor publications [41]–[45] for more details.

### C. MULTIMODAL RESTRICTED BOLTZMANN MACHINE (MRBM)

Extension of the RBM to the Multimodal Restricted Boltzmann Machine (MRBM) has been proposed in some studies [46]. MRBM is like two distinct RBMs which tied together

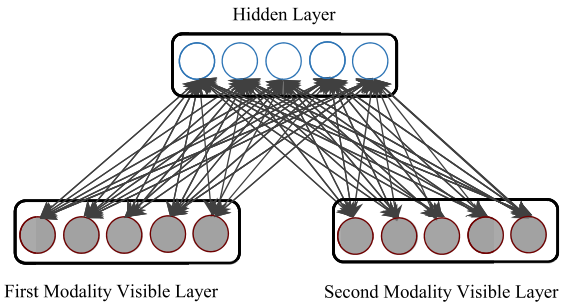


**FIGURE 5.** Multimodal restricted boltzmann machine (MRBM).

in their hidden layers (Fig. 5). The hidden layer is treated as the joint representation across modalities and the joint distribution over the hidden layer and the visible layers is defined according to (18):

$$p\left(v^{m_1}, v^{m_2}, h\right) = \frac{1}{Z}\exp\left(-E\left(v^{m_1}, v^{m_2}, h\right)\right) \tag{18}$$

$$\begin{aligned} E\left(v, h\right) = &-\sum_i a_i^{m_1} v_i^{m_1} - \sum_j a_j^{m_2} v_j^{m_2} - \sum_k b_k h_k \\ &- \sum_{ik} v_i^{m_1} w_{ik}^{m_1} h_k - \sum_{jk} v_j^{m_2} w_{jk}^{m_2} h_k \end{aligned} \tag{19}$$

$$\begin{aligned} p\left(v^{m_1}, v^{m_2}\right) &= \frac{1}{Z}\sum_h p\left(v^{m_1}, v^{m_2}, h\right) \\ &= 1/Z\sum_h \exp\left(-E\left(v^{m_1}, v^{m_2}, h\right)\right), \end{aligned} \tag{20}$$

where $Z$ is the normalizing constant, $v^{m_1}$, $v^{m_2}$ and $h$ are the binary visible variables assigned to the input modalities and binary hidden variables , respectively. $w^{m_1}$ and $w^{m_2}$ are the weights between each input modality and the hidden layer and $a^{m_1}$ and $a^{m_2}$ are the bias vectors corresponding to each modality.

Due to the bipartite structure, the variables in the same layer are conditionally independent given the variables of the other layers. As a result, the conditional probabilities could be written as follows:

$$p\left(h_k = 1 \mid v^{m_1}, v^{m_2}\right) = \sigma(b_k + \sum_i v_i^{m_1} w_{ik}^{m_1} + \sum_j v_j^{m_2} w_{jk}^{m_2}) \tag{21}$$

$$p\left(v_i^{m_1} = 1 \mid h\right) = \sigma(a_i^{m_1} + \sum_k w_{ik}^{m_1} h_k) \tag{22}$$

$$p\left(v_j^{m_2} = 1 \mid h\right) = \sigma\left(a_j^{m_2} + \sum_k w_{jk}^{m_2} h_k\right). \tag{23}$$

Variation of information (VI) or shared information distance [7], [8] is a metric that measures how much two random variables are informative about each other. In other words, VI is the total amount of uncertainty remaining about variables after the other one is known. It is defined as the addition of conditional entropies of two variables according to (25).

$$VI\left(X \cdot Y\right) = \left[H\left(X\right) - I\left(X \cdot Y\right)\right] + \left[H\left(Y\right) + I(X \cdot Y)\right] \tag{24}$$

$$VI\left(X \cdot Y\right) = H\left(X \mid Y\right) + H\left(Y \mid X\right), \tag{25}$$

where $H$ and $I$ are used to denote the entropy and mutual information of two random variables named $X$ and $Y$. Using the conditional entropy formula, (25) can be rewritten as (27).

$$H\left(X \mid Y\right) = E_{p(X \cdot Y)}[\log p\left(X|Y\right)] \tag{26}$$

$$VI = E_{p(X \cdot Y)}\left[\log p\left(X|Y\right) + \log p\left(Y|X\right)\right], \tag{27}$$

where conditional entropy of $Y$ given $X$ can be written in a similar way, $p\left(X \cdot Y\right)$ is the joint probability, $p\left(Y|X\right)$ and $p\left(X|Y\right)$ are the conditional probabilities of variables.

Sohn *et al.* [19] considered data modalities as random variables and defined new multimodal learning criteria, namely minimum variation of information learning (MinVI) as follows:

$$\begin{aligned} MinVI : \mathcal{L}^{VI}\left(\theta\right) : &\min_\theta \mathcal{L}^{VI}\left(\theta\right) \\ &= -E_{p_D(X \cdot Y)}\left[\log p_\theta\left(X|Y\right) + \log p_\theta\left(Y|X\right)\right], \end{aligned} \tag{28}$$

where, $p_\theta$ is any distribution on random variables which is parametrized by $\theta$ and $p_D$ denotes the estimated probabilities from data. It is shown that MinVI objective can be decomposed into a sum of two negative conditional log likelihoods. Sohn *et al.* [19] provided a theoretical proof of why the proposed learning objective is sufficient to estimate the joint distribution of the multimodal data.

Taking a look at conventional NLL criteria, it includes four KL divergence terms including two marginal and two conditional distributions. Due to the greater number of modes in the marginal distributions compared with the conditional distributions, Sohn *et al.* [19] reasoned that KL divergences of marginal distributions may become a dominant factor during the minimization process. Thus, it prevents the model from learning a good association between data modalities. In fact, models with NLL training criteria try to learn whole data distribution and in turn, MinVI objective focuses on modeling the conditional distributions of data modality which is arguably easier to minimize and will result in learning much more informative cross-modal representations.

$$\begin{aligned} ML : \min_\theta \mathcal{L}^{NLL}\left(\theta\right) : \mathcal{L}^{NLL}\left(\theta\right) \\ = -E_{p_D(X \cdot Y)}\left[\log p_\theta\left(X \cdot Y\right)\right] \end{aligned} \tag{29}$$

$$\begin{aligned} \mathcal{L}^{NLL}\left(\theta\right) = &\frac{1}{2}\left(KL(p_D(X)||p_\theta\left(X\right)) + KL(p_D\left(Y\right)||p_\theta\left(Y\right)\right) \\ &+ E_{p_D(Y)}\left[KL(p_D(Y|X)||p_\theta(Y|X)\right] \\ &+ E_{p_D(Y)}\left[KL(p_D(X|Y)||p_\theta\left(X \mid Y\right)\right]\right) + C \end{aligned} \tag{30}$$

$$\begin{aligned} \mathcal{L}^{NLL} = &\frac{1}{2}\Big(\mathcal{L}^{VI}\left(\theta\right) + \big(E_{p_D(Y)}\left[KL(p_D(Y|X)||p_\theta(Y|X)\right] \\ &+ E_{p_D(Y)}\left[KL(p_D(X|Y)||p_\theta\left(X \mid Y\right)\right]\big) + C\Big) \end{aligned} \tag{31}$$

A similar idea has been used for Generalized Denoising-Autoencoder (GDAE) [47] and Generative Stochastic Networks (GSNs) [48]. The intractable problem of learning whole data density is bypassed by focusing on learning the transition operators between clean and corrupted version of data or data and an arbitrary latent variable. Bengio *et al.* [47]

proved that the stationary density of GDAE converges to the original data distribution in the proposed successive noising and denoising process. Likewise, learning transition operators of GSNs is sufficient to learn a good generative model that estimates the data-generating distribution [48].

As expressed before, MRBM can be trained through minimizing the joint NLL using SGD while approximating the gradient with one member of CD family, but Sohn *et al.* [19] adapted two training methods to train MRBM with MinVI criteria including CD-PercLoss [49] and Multi-Prediction (MP) [50]. They stated that MP has a higher preference due to a few practical issues of the CD-PercLoss. MP computes the values of the hidden and the visible layers of MRBM via (21) to (23), where there is no missing modality and all required data are available in the training phase. Considering $v^{m_2}$ as the missing modality, the variational inference proceeds by alternately updating the mean-field parameters $\widehat{h}$ and $v^{m_2}$ which are initialized with all zeroes through the following equations:

$$\widehat{h}_k = \sigma(b_k + \sum_i v_i^{m_1} w_{ik}^{m_1} + \sum_j \widehat{V}_j^{m_2} w_{jk}^{m_2}) \tag{32}$$

$$\widehat{V}_j^{m_2} = \sigma\left(a_j^{m_2} + \sum_k w_{jk}^{m_2} \widehat{h}_k\right). \tag{33}$$

We refer the reader to the work of Sohn *et al.* [19] to see more details and proofs of theorems and declare that we use MRBM trained with MinVI in the top layer of our model.

## IV. PROPOSED MODEL

Although CDBN has the high modeling capability, it is proposed for modeling unimodal time series and it must be modified to account for multiple modalities jointly. So, in this paper, a new architecture named Multimodal Conditional Deep Belief Network (MCDBN) is proposed. Determining the structure, training and inference algorithms are required to specify a deep architecture completely. So in the remainder of this section, three aforementioned items are described in the same order as listed.

### A. MODEL STRUCTURE

As can be seen in the reviewed papers in Section II, most of the proposed multimodal deep structures choose the common strategy to learn a compact representation through the layers of modality-specific networks, firstly. Then the obtained representations are fused by a higher level network to learn a joint representation that is shared across multiple modalities. In this way, the learned representation has less within-modality correlation than the raw features and it is much easier for the fusion layer to model and capture between the modality relations. As it is depicted in Fig. 6, we utilize a similar bisectional structure that has a CDBN for each input time series in two lower layers and the top layer contains an MRBM. Applied CDBNs have two layers with the similar architecture. Since the motion data is a real-valued data, Gaussian CRBM is used in lower layers of CDBNs. The hidden units of Gaussian CRBM are binary so the conventional binary CRBM can be used in the upper layers.

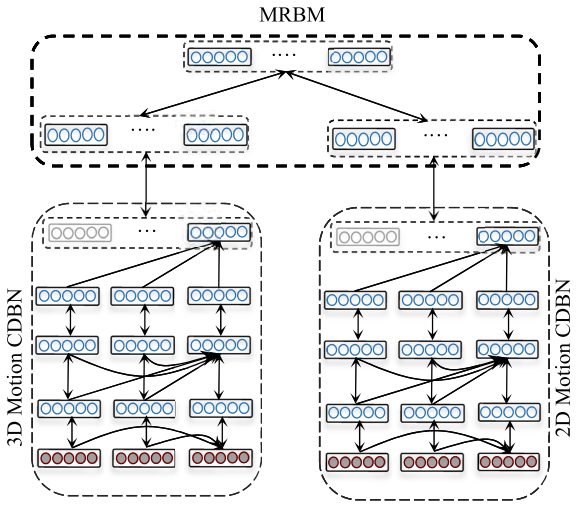


**FIGURE 6.** Multimodal conditional deep belief network (MCDBN).

In Fig. 6, fully connected links between layers in CDBNs and fully connected links between hidden and visible layers of MRBM are shown with bold arrows, for simplicity of visualization. So, each arrow between two layers either directed or bidirectional represents a set of links that connect all the variables of two layers together.

Multimodal motion datasets contain two synchronized 3 × n-dimensional time series of 3D motion data and 2 × n-dimensional time series of 2D motion data. Let $V^{3D}$ denotes the 3D motion data and $V^{2D}$ denotes the 2D motion data. MCDBN is able to regenerate 3D motion data, given 2D motion data via sampling from $p(V^{3D}|V^{2D})$. Due to the complexity and intractability of computing such a conditional density, we used the chain rule according to (35) and used a distinct architecture for learning each of the conditional density.

$$p\left(V^{3D} \mid V^{2D}\right) = p\left(V^{3D} \mid H^{3D}\right) p\left(H^{3D} \mid H^{2D}\right) p\left(H^{2D} \mid V^{2D}\right), \tag{34}$$

where $H^{3D}$, $H^{2D}$ and $H^{MRBM}$ denote the hidden layer of CDBN for 3D motion data, the hidden layer of CDBN for 2D motion data and the hidden layer of the MRBM, respectively.

### B. TRAINING

Equation 36 defines the training objective of MCDBN. Due to the independence of optimization terms in this equation, the CDBNs for 2D and 3D motion data and the MRBM are trained, separately.

$$ML : \min_{\theta_{MCDBN}} \mathcal{L}(\theta_{MCDBN}) : \mathcal{L}(\theta_{MCDBN}) = \frac{1}{2}\left[\mathcal{L}^{NLL}(\theta_{3DCDBN}) + \mathcal{L}^{NLL}(\theta_{2DCDBN}) + \mathcal{L}^{VI}(\theta_{MRBM})\right] \tag{35}$$

**Algorithm 1** Training Procedure of MCDBN Pseudocode

1: **MCDBN Training** ()
2: **input**: $V^{3D}$, $V^{2D}$
3: **output**: $\theta_{MCDBN} = [\theta_{3D\,CDBN}, \theta_{2D\,CDBN}, \theta_{MRBM}]$
4: $\left[H^{3D}, \theta_{3D\,CDBN}\right]$
$= Train\_3D\,motion\_CDBN(V^{3D}, NLL)$
5: $\left[H^{2D}, \theta_{2D\,CDBN}\right]$
$= Train\_2D\,motion\_CDBN(V^{2D}, NLL)$
6: $\left[H^{MRBM}, \theta_{MRBM}\right] = Train\_MRBM(H^{3D}, H^{2D}, VI)$
7: **end MCDBN Training**

**Algorithm 2** CD-K Pseudocode [35]

1: **CD_K** ()
2: **input**: $K$, Training data $D$
3: ***for*** $s \leftarrow 1$ *to size of Traininig data* **(D)**
4: ***for*** $k \leftarrow 1$ *to* $K$
5: Select the $s_{th}$ train data and consider it as a visible variables
6: Update all the hidden variables in parallel
7: Update all the visible variables in parallel to get reconstructed data
8: Update all the hidden variables again
9: Update all the hidden variables
10: ***end*** *for*
11: Update all the parameters (weights and biases)
12: ***end*** *for*
13: **end CD_K**

where $\theta_{3DC\,DBN}$, $\theta_{2D\,CDBN}$, and $\theta_{MRBM}$ denote the parameters of CDBN for 3D motion data, the parameters of CDBN for 2D motion data and the parameters of MRBM, respectively. The pseudocode of MCDBN training procedure is mentioned in Algorithm 1. The training procedure has three main steps. In each step, the parameters of each unit are trained through the corresponding optimization. In the pseudocode, the training function of CDBN for 3D motion data and the training function of CDBN for 2D motion data are denoted by $Train\_3D\,motion\_CDBN$ and $Train_2D\,motion\_CDBN$, respectively. CDBNs are trained disjointedly to learn conditional probabilities $p(H^{3D}|V^{3D})$ and $p(H^{2D}|V^{2D})$.

In training CDBNs, we used the same greedy layer-wise algorithm which is proposed by Taylor [41]. In more details, for training a CDBN Each individual CRBM either Gaussian or binary is trained by CD, initializing weights by random values drawn from the standard normal distribution. The parameters of lower CRBMs are frozen while training upper layers and the sequence of hidden variables driven by the data are treated as new visible data for training upper CRBMs. This greedy learning algorithm is guaranteed to never decrease in a variational lower bound on the log probability of the data under the full generative model [38].

CD is based on a maximum likelihood learning rule that is mentioned in (36):

$$\Delta\theta_{CDBN} \propto \langle \frac{\partial \mathcal{L}^{NLL}(\theta_{CDBN})}{\theta_{CDBN}} \rangle_{data} - \langle \frac{\partial \mathcal{L}^{NLL}(\theta_{CDBN})}{\theta_{CDBN}} \rangle_{rec} \quad (36)$$

where $\langle\cdot\rangle_{data}$ is the expectation with respect to the real data distribution and $\langle\cdot\rangle_{rec}$ is the expectation with respect to the reconstructed data, starting from a data vector on the visible units and Gibbs sampling between all the hidden and all the visible variables, $K$ times. The pseudocode of CD algorithm is mentioned in Algorithm 2.

The update rule mentioned in line 11 of Algorithm 2 is done using (37) to (41). These update rules consider the effects of the previous visible variables on the current hidden units. The updates for the directed weights are also based on simple pairwise products.

$$\Delta W_{ij} \propto \langle v_i^t h_j^t \rangle_{data} - \langle v_i^t h_j^t \rangle_{rec} \quad (37)$$
$$\Delta A_{ki} \propto \langle v_i^t H_k^t \rangle_{data} - \langle v_i^t H_k^t \rangle_{rec} \quad (38)$$
$$\Delta B_{kj} \propto \langle h_j^t H_k^t \rangle_{data} - \langle h_j^t H_k^t \rangle_{rec} \quad (39)$$
$$\Delta a_i \propto \langle v_i^t \rangle_{data} - \langle v_i^t \rangle_{rec} \quad (40)$$
$$\Delta b_j \propto \langle h_j^t \rangle_{data} - \langle h_j^t \rangle_{rec} \quad (41)$$

where $v_i^t$ and $a_i$ are the $i_{\text{th}}$ current visible variable and its dynamic bias, $h_j^t$ and $\mathrm{b_j}$ are the $j_{\text{th}}$ current hidden variable and its dynamic bias, respectively. $H_k^t$ is the $k_{\text{th}}$ visible variable at history and $A_{ki}$ and $B_{kj}$ are the weights of connection from the $k_{\text{th}}$ visible variable at history to the $i_{\text{th}}$ current visible and the $j_{\text{th}}$ current hidden variables, respectively. While learning a single CRBM, there is no need to proceed sequentially through the training data sequences. The updates are only conditional on the previous $N$ (the order of CRBM) time steps. So, each of $N+1$ frames are mixed into a mini-batch.

After training CDBNs, the values of hidden variables produced from 3D motion data ($H^{3D}$) and the values of hidden variables produced from 2D motion data ($H^{2D}$) are fed as inputs to the MRBM which handles the fusion task. In the pseudocode, the training function of MRBM is denoted by $Train\_MRBM$. Then, the parameters of the MRBM ($\theta_{MRBM}$) would be tuned through the following optimization (MinVI criteria).

$$\begin{aligned} MinVI : \min_{\theta} \mathcal{L}^{VI}(\theta_{MRBM}) :\ & \mathcal{L}^{VI}\left(\theta^{MRBM}\right) \\ = & -E_{p_D(H^{3D}.H^{2D})}\Big[\log p_{\theta^{MRBM}}\left(H^{3D}|H^{2D}\right) \\ & + \log p_{\theta^{MRBM}}\left(H^{2D}|H^{3D}\right)\Big], \end{aligned} \quad (42)$$

As declared in the Section III.c, the MRBM is trained via the MP algorithm that tries to find the parameters that minimize the above optimization using SGD while computing the gradient by back-propagating the error between the ground truth data and the predicted data via (32) and (33).

In contrast to the conventional approach in the deep learning community, our experiments showed that fine-tuning does not improve the results much considerably. It seems rational because in most cases, network fine tuning is done in case of

the end-to-end learnings but MCDBN consists of successive separate networks; so finally, there is no need to fine-tune the whole structure.

After the training procedure, the CDBNs would be able to extract the compact representation from raw data and also they have the ability to regenerate visible data given hidden layer values. Likewise, the MRBM learns the cross-modal relationship in the training phase and would be able to regenerate representation of 3Dmotion data given only 2D motion data. So, the MCDBN is able to regenerate data in the inference phase.

Considering the described training procedure in Algorithm 1, the computational complexity of training MCDBN can be calculated according to (43):

$$\begin{aligned} O\left(training_{MDBN}\right) = & O\left(Train\_2D\,motion\_CDBN\right) \\ & + O\left(Train\_3D\,motion\_CDBN\right) \\ & + O\left(Train\_MRBM\right). \end{aligned} \tag{43}$$

If the computational cost of one Gibbs transition is $T$, and the computational cost of evaluating variables is denoted by $L$, the computational cost of the CD-K algorithm for a dataset of size $n$ is $O(n(KT + 2L))$ [51]. In the proposed MCDBN, the computational cost of one Gibbs transition and also the cost of evaluating variables both are equal to $O(1)$, so the overall cost of utilized CD algorithm for training all CRBMs is $O(n)$. The training procedure of the MRBM with VI criterion is also based on CD and its training cost would be $O(n)$, too. Therefore, the total computational cost of training proposed MCDBN can be considered as O(n), where $n$ is the size of input data (number of frames).

### C. INFERENCE

After learning MCDBN parameters in the training phase, the MCDBN should be able to regenerate 3D motion data given only 2D motion data in the inference time. If the regenerated 3D motion is represented by $\widehat{V}^{3D}$, the inference equation will be as (44):

$$\begin{aligned} & p\left(\widehat{V}^{3D} \mid V^{2D}\right) \\ & \quad = p\left(\widehat{V}^{3D} \mid \widehat{H}^{3D}\right) p\left(\widehat{H}^{3D} \mid H^{2D}\right) p\left(H^{2D} \mid V^{2D}\right). \end{aligned} \tag{44}$$

The pseudocode of MCDBN inference procedure is mentioned in Algorithm 3. Similar to the training procedure, three actions must be done. First, the compact representation of 2D motion data ($H^{2D}$) is generated from 2D motion data ($V^{2D}$). Secondly, the compact representation for 3D motion data ($\widehat{H}^{3D}$) is produced by the top layer MRBM from the compact representation of 2D motion data ($H^{2D}$). To model $p\left(\widehat{V}^{3D} \mid \widehat{H}^{3D}\right)$, the estimated compact representation for 3D motion data ($\widehat{H}^{3D}$) will be forwarded to 3D motion CDBN to infer the visible 3D motion data $\widehat{V}^{3D}$, using it as the hidden states. In the pseudocode, the inference functions of CDBN for 2D motion data, the CDBN for 3D motion data and the MRBM are denoted by *Inference_2D motion_CDBN* and *Inference_3D motion_CDBN*, and *Inference_MRBM*, respectively.

**Algorithm 3** Pseudocode of Inference Procedure for MCDBN

1: **MCDBN Inference ()**
2: **input**: $V^{2D}$, $\theta_{MCDBN} = [\theta_{3D\,CDBN}, \theta_{2D\,CDBN}, \theta_{MRBM}]$
3: **output**: $\widehat{V}^{3D}$
4: $\left[H^{2D}\right] = Inference\_2D\ motion\_CDBN(V^{2D}, \theta_{2D\,CDBN})$
5: $\left[\widehat{H}^{3D}\right] = Inference\_MRBM(H^{2D}, \theta_{MRBM})$
6: $\left[\widehat{V}^{3D}\right] = Inference\_3D\ motion\_CDBN(\widehat{H}^{3D}, \theta_{3D\,CDBN})$
7: **end MCDBN Inference**

**Algorithm 4** Pseudocode of CDBN Inference Proposed by Taylor [41]

1: **CDBN Inference ()**
2: **inputs**: the first $N_1 + N_2$ frames of $V$, $\theta_{CDBN} = \theta^1_{CRBM}, \theta^2_{CRBM}], f_n$
3: **output**: $\widehat{V}$
4: Initialize the first $N_1 + N_2$ frames of $\widehat{V}$ with the corresponding frame in $V$ ($\left[V^1 : V^{N_1+N_2}\right] = [\widehat{V}^1 : \widehat{V}^{N_1+N_2}]$)
5: Initialize $N_2$ frames of the first hidden layer using a mean-field up-pass through the first CRBM
6: ***for*** $i \leftarrow 1$ *to* $f_n$
7: Initialize the hidden layer variables at $N_1 + N_2 + i$ to the value of the hidden layer variables at $N_1+N_2+i-1$.
8: Perform an alternating Gibbs sampling in the 2nd layer CRBM.
9: Do a mean-field down-pass in the first layer CRBM to obtain the visible variables at time $N_1 + N_2 + i$ ($\widehat{V}^{N_1+N_2+i}$)
10: ***end*** *for*
11: **end CDBN Inference**

In contrast to learning procedure in which training both CDBNs is done through the proposed approach by Taylor, the inference is somehow different. More concretely, generating compact representation from 2D motion data is straightforward using the learned parameters. The top layer MRBM would also generate 3D motion compact representation through the manner described in Section III.c. But, inference procedure is slightly different from the approach proposed by Taylor for inferring in 3D CDBN. The inference procedure of a two-layer CDBN proposed by Taylor [41] is mentioned in Algorithm 4. It is assumed that the order of lower CRBM is $N_1$ and the order of upper CRBM is $N_2$. $\widehat{V}$ is the regenerated data, $f_n$ is the desired number of frames to regenerate, $\theta^1_{CRBM}$ and $\theta^2_{CRBM}$ denote the parameters of the lower CRBM and the parameters of the upper CRBM, respectively.

Our proposed inference procedure for 3D CDBN is mentioned in Algorithm 5, where $\widehat{H}^{3D}_i$ denote the hidden variables of $i_{\text{th}}$ frame and $\widehat{V}^{3D}_i$ denote the $i_{\text{th}}$ frame of 3D regenerated data. In the Taylor's approach, except the first $N_1+N_2$ frames, the hidden variables of next frames are initialized by copying

**Algorithm 5** Pseudocode of Proposed Inference Procedure for 3D CDBN

1: Inference_ 3D motion_CDBN()
2: **input**: $\widehat{H}^{3D}$, $\theta_{3D\,CDBN}$
3: **output**: $\widehat{V}^{3D}$
4: ***for*** $i \leftarrow 1$ *to size* $(\widehat{H}_i^{3D})$
5: Initialize the hidden layer variables at the top-most layer to $\widehat{H}_i^{3D}$.
6: Perform a layer-wise alternating Gibbs sampling using $\theta_{3D\,CDBN}$ parameters.
7: Do a mean-field down-pass to obtain the visible variables $\widehat{V}_i^{3D}$.
8: ***end*** *for*
9: **end** Inference_3D motion_CDBN

the previous time step hidden variables. But, in our approach, the hidden variables of all time frames are available in the 3D compact representation which are inferred by MRBM. So, there is no need to initialize them by copying previous time steps and hidden states of upper CRBM are initialized by MRBM produced data. Then an alternative Gibbs sampling will be done to generate the visible 3D motion data.

Considering the inference procedure described in Algorithm 3, the computational complexity of MCDBN inference can be calculated according to (45):

$$\begin{aligned} O\left(training_{MDBN}\right) = {} & O\left(Inference_2 Dmotion\right) \\ & + O\left(Inference_M RBM\right) \\ & + O\left(Inference_3 Dmotion\right), \end{aligned} \quad (45)$$

All the inference functions for 2D CDBN, 3D CDBN, and the MRBM are used Gibbs transitions to obtain the hidden and visible variables. Since the computational cost of Gibbs transition is $O(1)$, the computational cost of all three mentioned networks are $O(1)$, and the computational complexity of inference procedure for MCDBN is $O(1)$.

## V. EVALUATION

In this section, we investigate the performance of the proposed model on two datasets. The datasets, the pre-processing and the post-processing steps are described firstly. Then, the evaluation metric is discussed and our results are presented.

### A. DATASETS

We focus our analysis on Berkeley Multimodal Human Action Database (Berkeley MHAD) [52] and Carnegie Mellon University motion capture (CMU Mocap) [53] datasets, which are numbered among standard benchmarks. Berkeley MHAD [52] dataset is one of the most complete available datasets. It contains about 1 Terabyte of information. 12 subjects do 11 actions in 5 repetitions. Five synchronal distinct systems including one optical motion capture system, four multi-view stereo vision camera arrays, two Microsoft Kinect cameras, six wireless accelerometers, and four microphones were mounted to capture the performed actions that are done in a 2m $\times$ 2m square. Since the dataset provides camera parameters, 2D motion data on the image plains easily can be obtained from their corresponding 3D motion data.

CMU Mocap [53] dataset is one of the well-known datasets. It contains a vast variety of movements, performed by different subjects and captured from different point of views. CMU Mocap provides synchronized images and motion capture data. Videos are captured by 12 Vicon infrared MX-40 cameras mounted around a 3m $\times$ 8m area. The dataset contains both marker positions and skeleton movements. We used the skeleton movements for jumping, forward jump and walking actions. Unfortunately, this dataset does not provide camera parameters, so we assume a camera view with known parameters and generate 2D motion data from 3D motions synthetically.

We utilized these two datasets, due to their completeness and the diverse type of actions they contain. So, acceptable performance on these datasets could provide confidence about the results and show the robustness of MCDBN to variations in viewpoint, subject's anthropometric features and other factors such as rapidness of movement.

The bio-vision hierarchy or BVH file format is a popular way to store and manipulate human motion data. A BVH file contains ASCII text in which the first part of it stores the initial pose specifications of a human skeleton and specifications for subsequent poses which are provided in the remaining. We assume that the input 3D motion data are in BVH format. Berkeley MHAD contains BVH files, but CMU Mocap is in a different format. So, we downloaded and used CMU dataset in BVH format from [54]. The data are mean subtracted and are divided by the standard deviation to get a normalized data before training.

The regenerated 3D outputs motions are also in BVH format. It is required to convert regenerated BVH signals to 3D joint positions for playing and comparison purposes. So, a post-processing step has taken on these signals to generate the joints positions. Selecting coordinates systems will have no effect on the final results because the regenerated trajectories can be transformed into any arbitrary coordinate system through appropriate translations.

### B. EVALUATION METRIC

Many evaluation metrics have been introduced for 2D and 3D pose estimations. However, finding a suitable evaluation metric for generative models is an active area of research [25]. The previously proposed evaluation metrics can be categorized into two main groups. The metrics of the first group that are usually used in computer vision literature, are measurements based on the Euclidean distance of the estimated and the ground-truth joint positions in data frames. Probability of Correct Key-Points (PCK) [55], Probability of Correct Pose (PCP) [56], whole body Root Mean Square Error (RMSE) [3] and Mean Per Joint Position Error (MPJPE) [57] are samples of these metrics. The measurements of the second group are introduced for evaluating the generative models [58].

They are commonly based on Log-likelihood or equivalently Kullback-Leibler divergence.

As declared earlier in this paper, our main purpose is to provide a learning model which can regenerate similar not but exact 3D movements given 2D motion data. So, we need to apply a metric in which its evaluation would be via resemblance of the generated and the original data. Furthermore, if an action is performed by different subjects or when an individual does the action at different times, the motion trajectories are not exactly the same, but they are correlated. So, we have to check how all the regenerated motions are similar and check it against correlations of real motion signals to verify the convergence properties. Therefore, the regenerated motions should be investigated from two points of view. First, how much the 3D regenerated motions are correlated with the ground truth 3D motions and second, the similarity of whole regenerated motions with each other and comparing it against the similarity of real motions done by humans.

The correlation coefficient is a metric for measuring the dependence of two random variables. One type of correlation coefficient developed by Karl Pearson is named Pearson correlation coefficient (PCC), also referred to as the Pearson's $r$, Pearson product-moment correlation coefficient (PPMCC) or bivariate correlation [59], [60]. When PCC is applied to a population, usually it is represented by the Greek letter $\rho$ and referred to as the population Pearson correlation coefficient. The PCC measures the strength and direction of the linear relationship between two random variables as the covariance of the two variables divided by the product of their standard deviations as follows:

$$\rho_{XY} = \left( \frac{cov(X, Y)}{\sigma_X \sigma_Y} \right). \tag{46}$$

where $cov(X, Y)$, $\sigma_X$ and $\sigma_Y$ are covariance and standard deviations of two random variables $X$ and $Y$, respectively. Applying Pearson's correlation coefficient to a sample would result in a measurement named as sample correlation coefficient or the sample Pearson correlation coefficient. It is usually represented by the letter $r$ and it can be obtained by substituting estimations into the above formula. So, if each random variable has $N$ scalar observations, the PCC is defined as the sample covariance of the variables divided by the product of their sample standard deviation according to (48).

$$r_{XY} = \frac{1}{N-1} \sum_{i=1}^{N} \left( \frac{X_i - \mu_X}{s_X} \right) \left( \frac{Y_i - \mu_Y}{s_Y} \right) \tag{47}$$

$$r_{XY} = \left( \frac{\sum X_i Y_i - N \mu_X \mu_Y}{(N-1)\, s_X s_Y} \right), \tag{48}$$

where $X_i$ and $Y_i$ are the single samples indexed with $i$. The symbols $\mu$ and $s$ denote the sample means and sample standard deviations, respectively. Both sample and population PCCs have the values between $+1$ and $-1$, where 1 is a total positive linear correlation, 0 is no linear correlation, and $-1$ is a total negative linear correlation. It is a symmetric measurement, so:

$$\rho_{XY} = \rho_{YX} \tag{49}$$

Another desired mathematical property is that PCC is invariant under the separate changes in location and scale of the two variables. This property is expressed in (50). Equation (50) shows that if variable $X$ is transformed to $a + bX$ and variable $Y$ is transformed to $c + dY$, where $a$, $b$, $c$, and $d$ are constants with $b, d > 0$, the population and sample PCC would not change. So, if the regenerated motion data differ from the ground truth data only in time shift and/or domain scale, this metric would not fail. In more details, in many cases where two real subjects do an action or a single subject repeat an action, motions visually look similar, but the subject starts the motion earlier or later. In other cases, the skeletal movements have similar trajectories with different domains. So, the distance metric should be robust to time shift and time scaling as well.

$$\rho_{XY} = \rho_{(a+bX)(c+dY)} \tag{50}$$

$$r_{XY} = r_{(a+bX)(c+dY)}. \tag{51}$$

To investigate the proposed architecture, the PCCs of ground truth 3D motions and each of the regenerated ones are computed. Then, a histogram of these PCCs is plotted and statistics of PCCs are reported for each action. To check the second property, PCCs of every pair of real motion samples are computed. Then PCC histograms of randomly chosen pairs of regenerated movements against PCC histograms of real movements are sketched for the comparison.

Then, both real and regenerated data histograms were checked for the best fitting distributions in terms of AIC and BIC measurements. In most histograms, Generalized Extreme Value (GEV) distribution was placed in the first rank; while in all cases, the shape parameters of the GEV distributions were negative. Since a GEV distribution with negative shape parameter corresponds to Weibull families. We fit Weibull distributions to histograms and report the fitted parameters and the Kullback-Leibler (KL) divergence of these distributions.

The probability density function (PDF) of Weibull distribution is defined for $x \in [0.\infty)$ as:

$$f\,(x; k \cdot l) = \left( \frac{k}{l} \right) \left( \frac{x}{l} \right)^{k-1} e^{\left[ -\left( \frac{x}{l} \right)^{k} \right]}, \tag{52}$$

where $k > 0$ and $l > 0$ are the shape and scale parameters, respectively. The Weibull shape parameter is also known as the slope parameter because its value is equal to the slope of the regressed line in the probability plot. Different values of the shape parameter affect the behavior of the distribution, obviously. Variations in the scale parameter will change the abscissa scale. Increasing the value of $k$ would stretch out the PDF while holding $l$ constant and due to the constancy of area under the PDF, the peak value of the PDF curve will decrease with the increase in $k$. If $k$ is increased while $l$ is kept fixed, the distribution curve will stretch out to the right and

its height will decrease. If $k$ is decreased while $l$ is constant, the distribution curve gets pushed in towards the left and its height increases.

Due to the exponential form of Weibull distributions, KL divergence of two Weibull distributions has closed form and has been computed by Bauckhage [61]. According to the Bauckhage computation, the KL divergence between two Weibull densities $f\ (x; k1.l1)$ and $f\ (x; k2.l2)$ amounts to:

$$KL(f1||) = \ln\left(\frac{k_1}{l_1^{k1}}\right) - \ln\left(\frac{k_2}{l_2^{k_2}}\right) + (k_1 - k_2)\left[\ln(l_1) - \frac{\gamma}{k_1}\right] + \left(\frac{l_1}{l_2}\right)^{k_2} \times \Gamma\left(\frac{k_2}{k_1} - 1\right) - 1 \quad (53)$$

where $\Gamma$ is Gamma function and $\gamma \approx 0.5772$ is the Euler-Mascheroni constant. In fact, performing actions by humans or regenerating actions is like generating a sample from these distributions. So whatever these distributions have the lower KL divergence, the generated motions are more similar and seem more realistic.

### C. RESULTS

We evaluate our method on six motion categories of Berkeley MHAD dataset including Jumping in place, jumping-jacks, waving two hands, waving one hand, clapping hands and throwing a ball. Moreover, the proposed method is evaluated on three actions of CMU Mocap dataset, including jumping, walking, and forward jump. The specification of hardware and software utilized for the experiments is described in Table 1.

**TABLE 1. Hardware/software specification.**

| Hardware/Software | Values |
|---|---|
| CPU | Core i7 |
| Memory | 8 GIG DDR3 |
| Cache | 6 MB |
| GPU | NVidia GeForce GTX |
| O.S. | Windows 10 |
| Programming Language | MATLAB 2017.b |

The MCDBN has two types of parameters. The first group contains parameters that are used in the training and inference procedures and the other type of parameters corresponds to the architecture of MCDBN. Table 2 summarizes the leaning and inference parameters. We investigated the effect of parameter $k$ in both training and inference procedures. This parameter controls the number of Gibbs transitions in CD-k algorithm. Increasing $k$ would result in performance rise, as well as increasing the training and inference time. According to our results setting $k = 5$ and $k = 3$ lead to the best tradeoff between time and performance in training and inference procedures, respectively. We initialize the learning rate, momentum and batch size parameters inspiring from previous works [19] and tune them through the validation procedure.

**TABLE 2. MCDBN parameters.**

| Parameter | Value |
|---|---|
| Learning rate | 0.1 |
| Momentum | 0.9 |
| Batch size | 100 |
| $k$(Gibbs Sampling for training) | 5 |
| $k$(Gibbs Sampling for inference) | 3 |

There are two architectural parameters for MCDBN including the model order and the number of hidden variables. We also investigated the effects of these parameters via the validation procedure. Although, increasing the number of hidden variables and the order of CRBMs increase the performance, but they cause prolonging the training and inference procedures. So, there is a tradeoff between training and inference time and performance based on the number of hidden variables and model order. We tried different architectures with different orders and different number of hidden variables. The results are depicted in Fig. 7. In this graph, we encode the MCDBN architecture as a triplet where the elements show the order of CRBMs, number of hidden variables of CRBMs and the number of hidden variables of the MRBM, respectively. The vertical axis of this figure shows the mean value of PCCs gained by the mentioned architecture. The horizontal axes show the training and inference time. The winner architecture is depicted in red color in Fig. 7. It has 300 hidden variables in all CRBMs with order 5 and it has 150 hidden variables in the MRBM. The MCDBN with this architecture was successful in the trade-off between time and performance.

As mentioned before, every action in Berkeley MHAD dataset was performed by 10 subjects in 5 repetitions. So, the dataset contains 50 distinct samples for each action. We have randomly picked 4 samples out of 50 samples of MHAD dataset as training data for each action. Since the model only tries to learn the conditional distribution of current frames given previous ones, this number of training samples is enough. A typical sample of MHAD dataset contains about 3000 frames on the average, so the train data contain about 12000 frames. Moreover, 1 sample of each action is used for validating parameters. Given 2D motion data of these samples, the 3D motion data are regenerated 10 times. So, we have 500 regenerated 3D motions. Due to the few numbers of training and validation samples in comparison with the number of test samples and considering that we regenerate each sample ten times, we have reported the results for all actions not only test samples to be more significant.

According to the described reasons in the Section V.B, PCC of ground truth 3D motions and every ten regenerated ones have been computed. The histograms of PCCs between the real movements and 500 regenerated movements for different actions are depicted in Fig. 8 and the statistics of PCCs are reported in Table 3. All the PCCs in all actions in Fig. 8 are

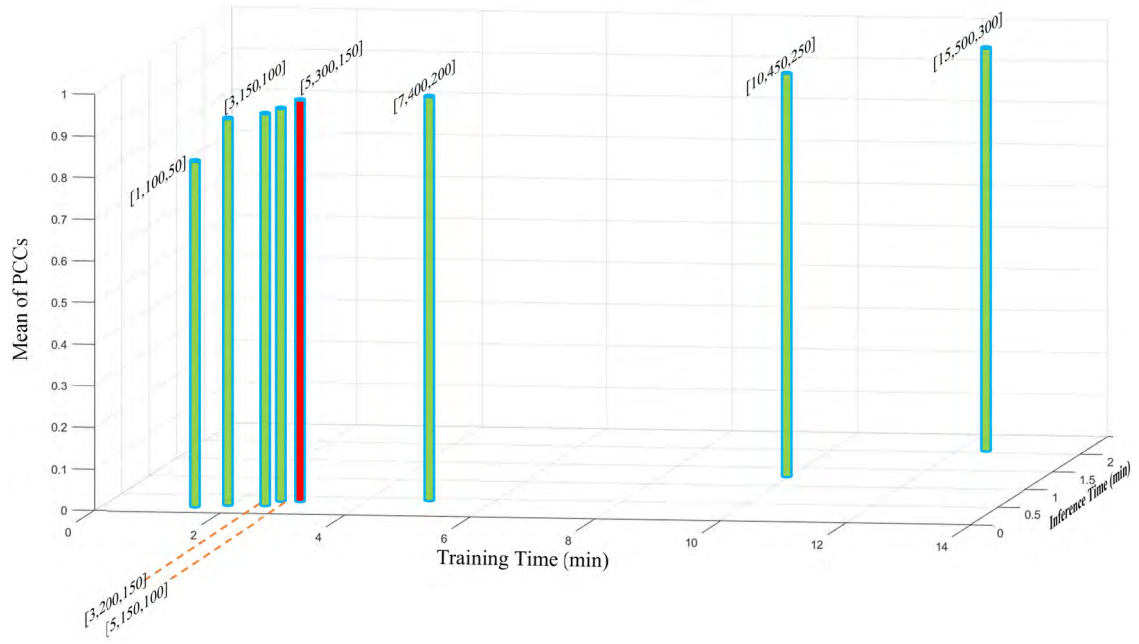


**FIGURE 7.** Performance, inference and training time of different architectures.

**TABLE 3.** Statistics of PCCs for regenerated actions of MHAD dataset.

| Action | Sample NO. | Mean | Standard Deviation | Mode | Median |
|---|---|---|---|---|---|
| Jumping in place | 500 | 0.9534 | 0.0288 | 0.8964 | 0.9604 |
| Jumping-jacks | 500 | 0.9436 | 0.0208 | 0.8507 | 0.9446 |
| Waving two hands | 500 | 0.9624 | 0.0213 | 0.9243 | 0.9683 |
| Waving one hand | 500 | 0.9547 | 0.1330 | 0.4432 | 0.9547 |
| Clapping hands | 500 | 0.9727 | 0.0282 | 0.8788 | 0.9682 |
| Throwing a ball | 500 | 0.9356 | 0.0275 | 0.8697 | 0.9412 |

greater than 0.85 that indicate a high correlation between the regenerated and the ground truth signals.

The minimum value of the PCCs in the histogram of jumping in place action (Fig. 8a) is about 0.89 and most of the PCCs are about 0.98. The mean PCC value is 0.9534 and the low standard deviation (0.288) shows the high confidence of results in different runs. The PCC histogram of real and regenerated movements for jumping-jacks action (Fig. 8b) shows the minimum value of PCC is about 0.88, the mean PCC value is 0.9436 and the standard deviation is (0.0208).

Fig. 8c shows the PCC histogram of real and regenerated movements for waving two hands action. The minimum value of PCCs is about 0.92. The PCC values are approximately equally distributed and one-third of PCCs are in the range of [0.98,0.99]. The mean PCC value is 0.9624 and low standard deviation (0.0213) shows the high confidence of results in different runs. The PCC histogram of real and regenerated movements for waving one hand action is plotted in Fig. 8d. Most of the PCC values are greater than 0.9 and near half of them are greater than 0.95. The mean PCC value is 0.9547 and although the standard deviation is higher than the other actions, it is about 0.13 which can be considered as a low value.

The minimum value of PCCs in the PCC histogram of real and regenerated movements for clapping hands (Fig. 8e) is about 0.88. The mean PCC value is 0.9727 and sixty percent of PCCs are in the range of [0.98, 1]. The histogram bin counts have an ascending order in the range of [0.87 1]. Fig. 8f shows the PCC histogram of real and regenerated movements for throwing a ball action. The PCC values are in the range of [0.86, 0.98], but most of the PCC values are greater than 0.88. The mean PCC value is 0.9356 and the standard deviation is 0.0275.

To inspect the convergence properties of regenerated movements, we have computed PCCs of every pair of 500 regenerated motion samples for all six actions of MHAD. It would result in 124750 PCCs. Histograms of PCCs for different actions are depicted in Fig. 9 and the parameters of fitted Weibull distributions to PCC histograms are reported in Table 4. We also report the significance levels for estimated parameters. The third and fourth columns of Table 4, give lower and upper bounds of 95% confidence

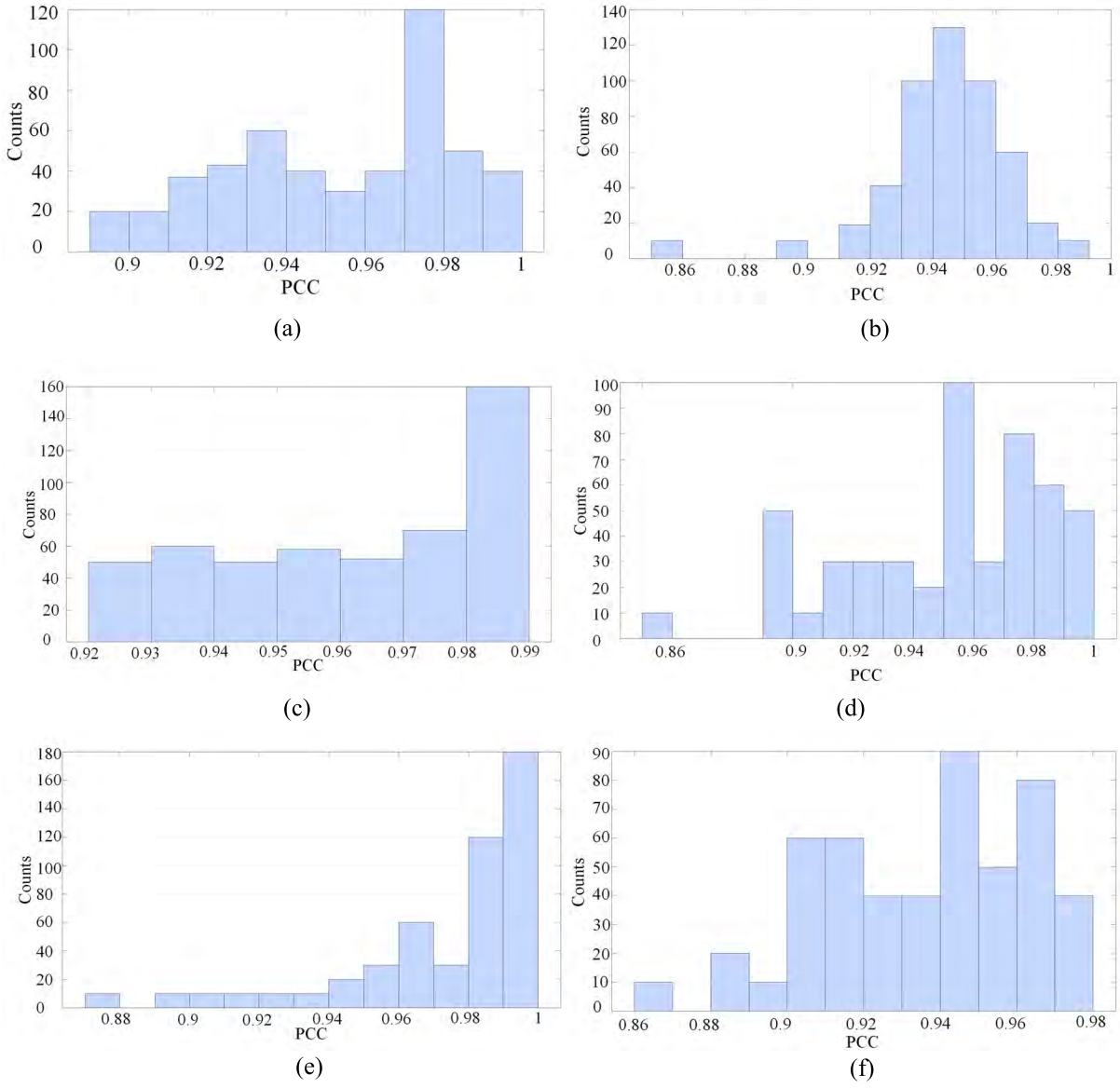


**FIGURE 8.** PCC histograms of regenerated actions of MHAD dataset a) jumping in place b) jumping-jacks c) waving two hands d) waving one hand e) clapping hands f) throwing a ball.

interval for shapes parameters and the fifth and sixth columns of Table 4, give lower and upper bounds of 95% confidence interval for scale parameters, respectively. Near all PCCs for all actions in Fig. 9 are greater than 0.98 and most of the PCCs are in the range of [0.99, 1]. It shows that there is a high correlation between regenerated 3D motions. Furthermore, the confidence intervals are tight that show the results are significant.

For each action, PCCs of every pair of 50 real motion samples have been computed which would result in 1225 PCCs. To compare the results fairly, the PCC histograms of 1225 randomly chosen pairs of regenerated movements against the PCC histograms of all real movements are depicted (Fig. 10). Two Weibull distributions are fitted to the histograms in red and blue colors for real and regenerated data, respectively. The parameters of the fitted distributions are reported in Table 5. The range of PCCs in Fig. 9 and Fig. 10 are very similar, that indicate sampling from PCCs is done unbiasedly and the sampled PCCs are distributed according to the main distributions in Fig. 9. However, the shape parameters in Table 4 tend to upper values due to more sample numbers.

The PCC histograms of pairwise real and regenerated motions for jumping in place action are very similar (Fig. 10a). The real data have lower mean PCC values in comparison with regenerated data. The PCC histograms of pairwise real and regenerated motions for jumping-jacks action are somehow similar (Fig. 10b). The real data have greater standard deviation and PCC values of real data are usually lower than the regener-

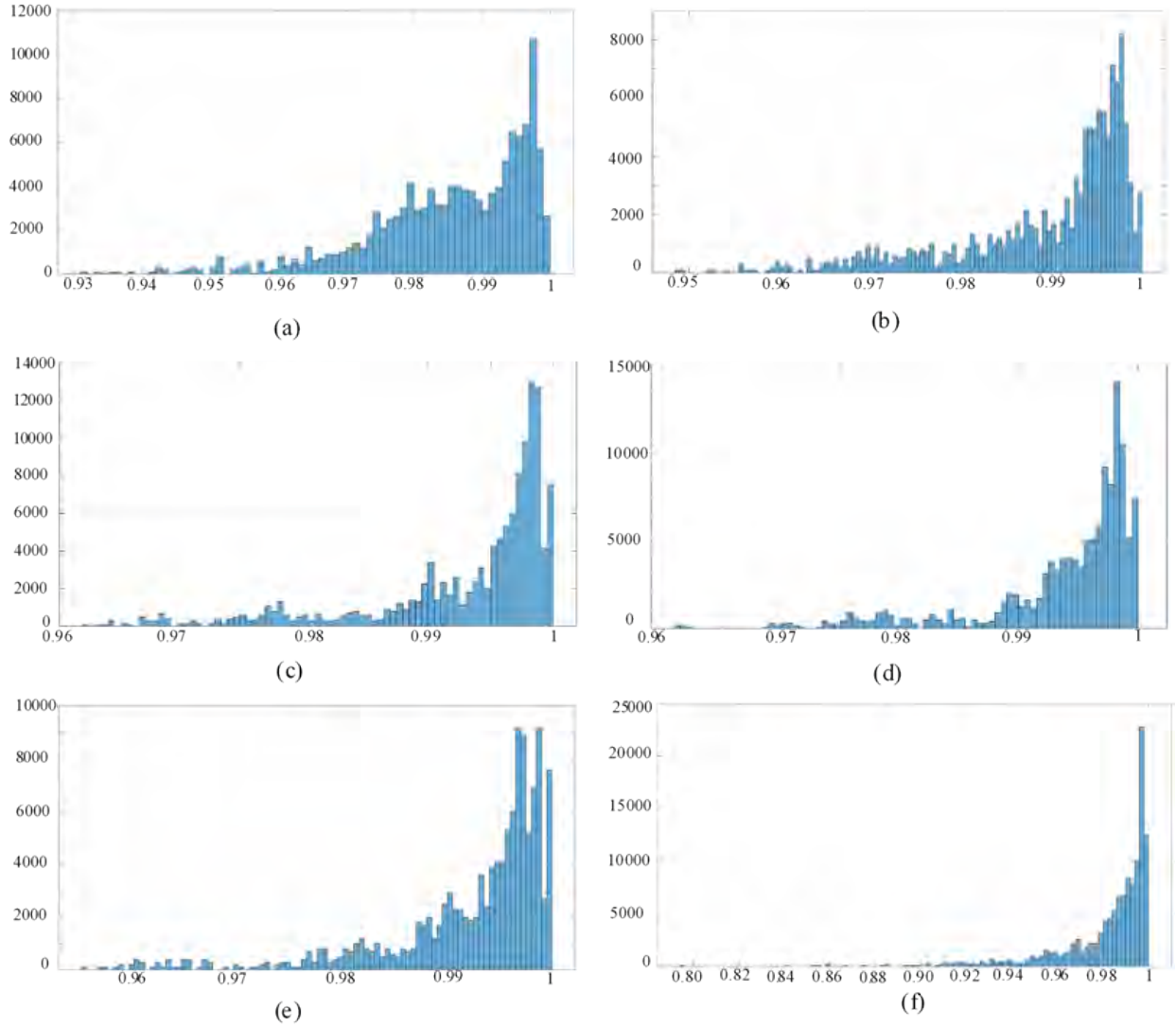


**FIGURE 9.** Pairwise PCC histograms of all regenerated actions of MHAD dataset a) jumping in place b) for jumping-jacks c) waving two hands d) waving one hand e) clapping hands f) throwing a ball.

**TABLE 4.** Parameters of fitted distributions to histograms of pairwise PCCs for all regenerated actions of MHAD dataset.

| Action | $k$ | $l$ | Lower bound $k$ | Upper bound $k$ | Lower bound $l$ | Upper bound $l$ |
|---|---|---|---|---|---|---|
| Jumping in place | 0.995076 | 116.650 | 0.995270 | 0.990626 | 116.114 | 117.188 |
| Jumping-jacks | 0.993691 | 158.621 | 0.993654 | 0.993727 | 157.858 | 159.388 |
| Waving two hands | 0.996181 | 223.805 | 0.996155 | 0.996206 | 222.789 | 224.998 |
| Waving one hand | 0.996430 | 258.820 | 0.996440 | 0.996452 | 257.566 | 260.790 |
| Clapping hands | 0.995482 | 206.547 | 0.995455 | 0.995510 | 205.555 | 207.453 |
| Throwing a ball | 0.990075 | 80.672 | 0.990005 | 0.990146 | 80.272 | 80.668 |

ated data which means the regenerated data have lower stochasticity.

The PCC histogram of pairwise real data in Fig. 10c has greater standard deviation and it contains lower PCCs. But it should be noted that, although the fitted distributions are not as similar as the other actions, their ranges are very close. The PCC values of real data are in the range of [0.92, 1] and the PCC values of regenerated data are in the range of [0.96, 1]. The PCC histogram of pairwise real data in Fig. 10d has greater standard deviation and it contains lower PCCs. Due to the similarity of this action with the previous action, the results are similar. The fitted distributions are not as

**TABLE 5.** Parameters of fitted distributions for real actions and samples of regenerated actions of MHAD dataset.

| Action | Real Data $k$ | Real Data $l$ | Regenerated Data $k$ | Regenerated Data $l$ | KL Divergence |
|---|---|---|---|---|---|
| Jumping in place | 0.9602 | 51.2917 | 0.9721 | 54.2110 | 0.3691 |
| Jumping-jacks | 0.9632 | 44.1413 | 0.9875 | 91.1867 | 1.3401 |
| Waving two hands | 0.9712 | 66.4063 | 0.9958 | 210.1931 | 4.9112 |
| Waving one hand | 0.9818 | 100.7294 | 0.9963 | 254.7183 | 2.4046 |
| Clapping hands | 0.9911 | 187.010 | 0.9941 | 165.7256 | 0.1091 |
| Throwing a ball | 0.9535 | 34.3108 | 0.9834 | 56.1873 | 0.8185 |

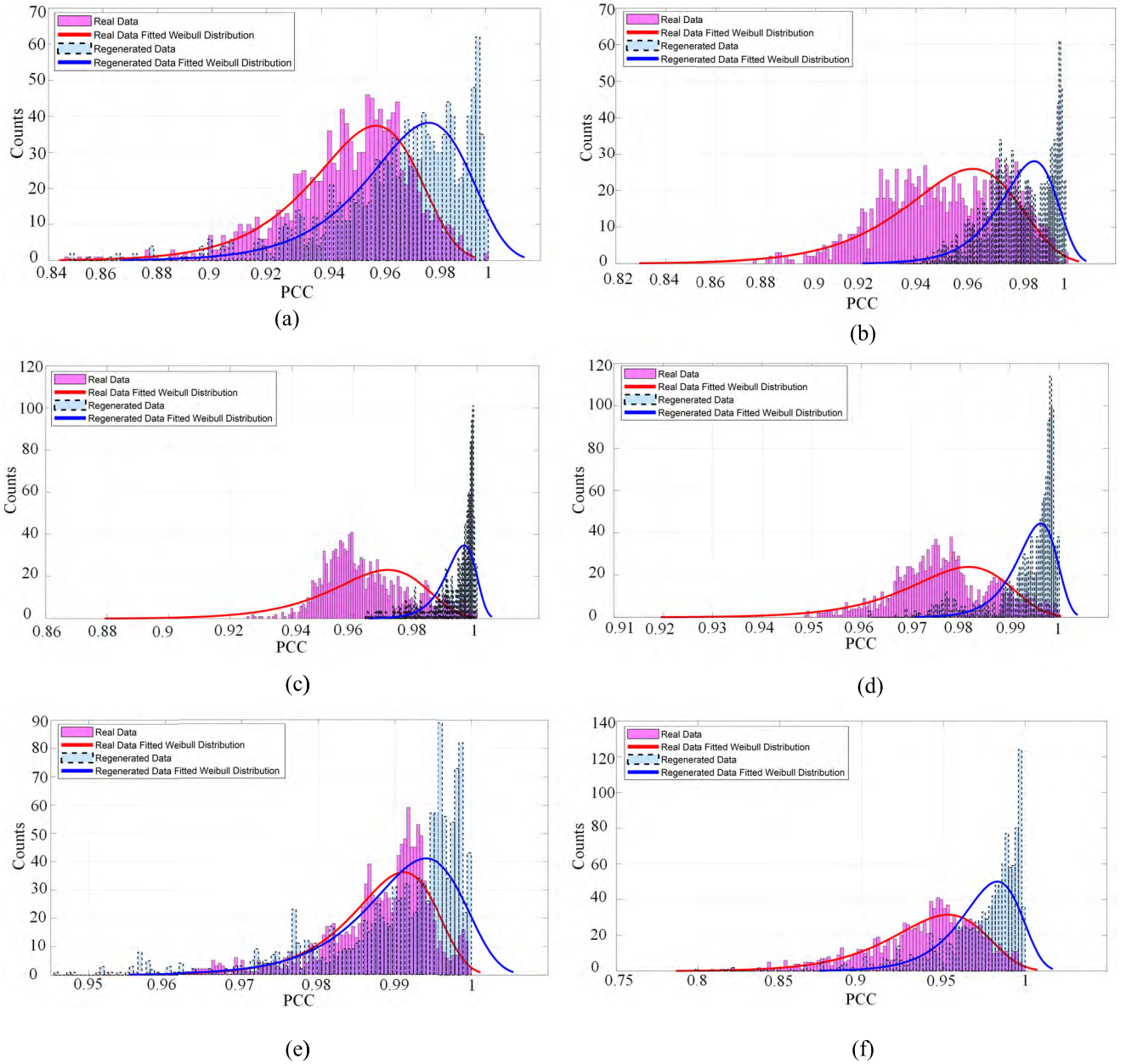


**FIGURE 10.** Pairwise PCC histograms of real and regenerated actions of MHAD dataset a) jumping in place b) for jumping-jacks c) waving two hands d) waving one hand e) clapping hands f) throwing a ball.

similar as the other actions. But it should be noted that the PCC values of real data are in the range of [0.951] and the PCC values of regenerated data are in the range of [0.971]. The fitted shape and scale parameters for these two actions in Table 5, obviously show the described situations. The PCC histogram of pairwise real and regenerated motions for

clapping hands action are very similar (Fig. 10e) and the parameters of fitted distributions are close to each other. The PCC histograms of pairwise real and regenerated motions for throwing a ball action is depicted in Fig. 10f. The real data has greater standard deviation and PCC values of real data are usually lower than regenerated data which means regenerated data has lower stochasticity.

Comparing jumping in place and jumping-jacks actions, we see that jumping in place action have more uniform distribution of PCCs in different bins of histograms (Fig. 8a and Fig. 8b). Furthermore, pairwise PCC histograms of real and regenerated motions (Fig. 10a and Fig. 10b) and the parameters of fitted Weibull distributions for these actions show that the regenerated motions for jumping in place best fit the real data. We believe that fewer employed joints in the jumping in place action causes this event.

The PCCs of regenerated motions in both waving two hands and waving one hand actions are nearer to one compared with real motions. It seems the difference could be justified as humans frequently do the small movement in joints of other parts of the body in addition to hands while doing this action, but the regenerated motions contain movements only in the joints of hands. The KL divergence of fitted Weibull distributions shows that the regenerated motions of waving one hand action are fitted better to real data. The reason could be the lower number of used joints in this action that simplifies the learning process.

The mean of PCCs for real and regenerated movements of clapping hands is the greatest mean compared with other PCC means. Likewise, the KL divergence of the real and regenerated motions has the lowest value among KL divergence of all actions. We believe that the uncomplicatedness of the clapping action, few numbers of engaged joints and symmetric joints movements in both sides of body simplifies the learning process and causes this superiority. The mean of PCCs of real and regenerated movements for throwing a ball action is the minimum value among all PCC means. It can be seen in pairwise PCC histogram of real and regenerated motions of throwing a ball action that the PCC values of real motions have greater range. It means the action is harder to learn due to variations in real data. So, the results are reasonable.

A similar procedure is performed for CMU dataset. The only difference is that the number of samples in CMU Mocap are not equal for different actions. There are 6 samples for jumping, 9 and 81 samples for forward jump and walking, respectively. The histograms of PCCs between real and regenerated movements of CMU Mocap dataset are depicted in Fig. 11 and the statistics of results are reported in Table 6.

The pairwise PCC histogram of real and regenerated motions for CMU Mocap dataset are depicted in Fig. 12 and the parameters of fitted distributions are reported in Table 7.

Fig. 11a and Fig. 12a show the PCC histogram of real and 60 regenerated motions and the PCC histograms of pairwise real and regenerated motions for jumping action of CMU dataset, respectively. All the PCC values are in the narrow range of [0.950.99]. The mean PCC value is 0.9662 and the low standard deviation (0.0161) show high confidence of the results in different runs. The histograms are very similar and the fitted parameters are also close to each other.

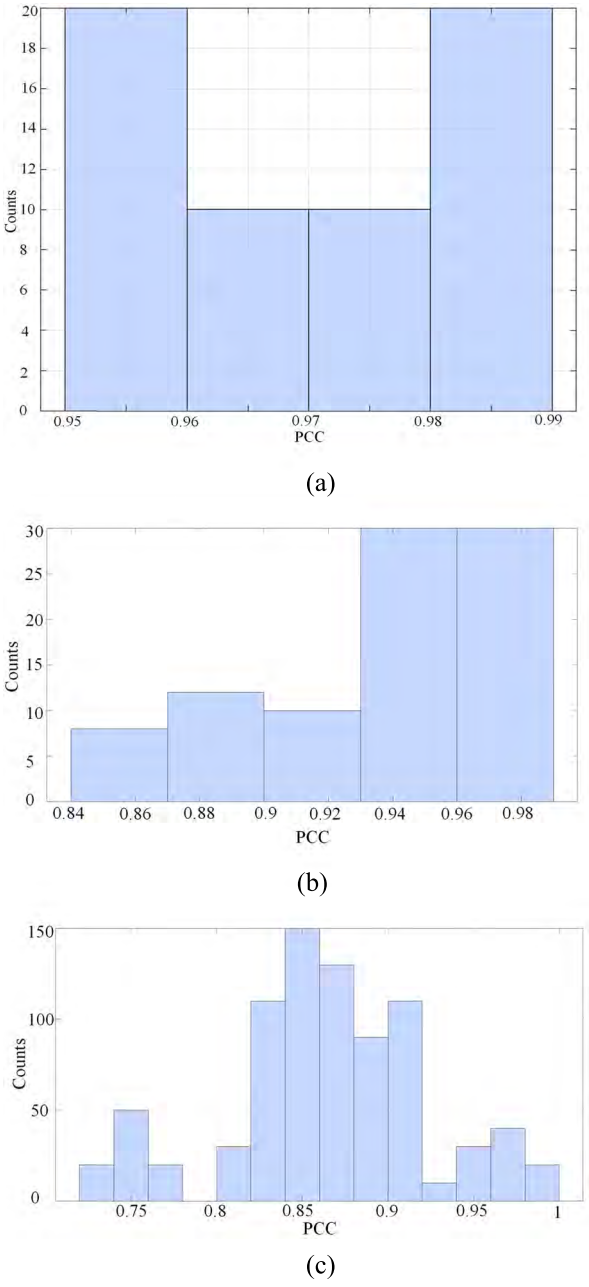


**FIGURE 11.** PCC histograms of regenerated actions of CMU Mocap dataset a) jumping b) forward jump c) walking.

**TABLE 6.** Statistics of PCCs for regenerated actions of CMU Mocap dataset.

| Action | Sample NO. | Mean | Standard Deviation | Mode | Median |
|---|---|---|---|---|---|
| Jumping | 60 | 0.9662 | 0.0161 | 0.9509 | 0.9662 |
| Forward jump | 90 | 0.9387 | 0.0396 | 0.9358 | 0.86803 |
| Walking | 810 | 0.8646 | 0.0574 | 0.9350 | 0.9600 |

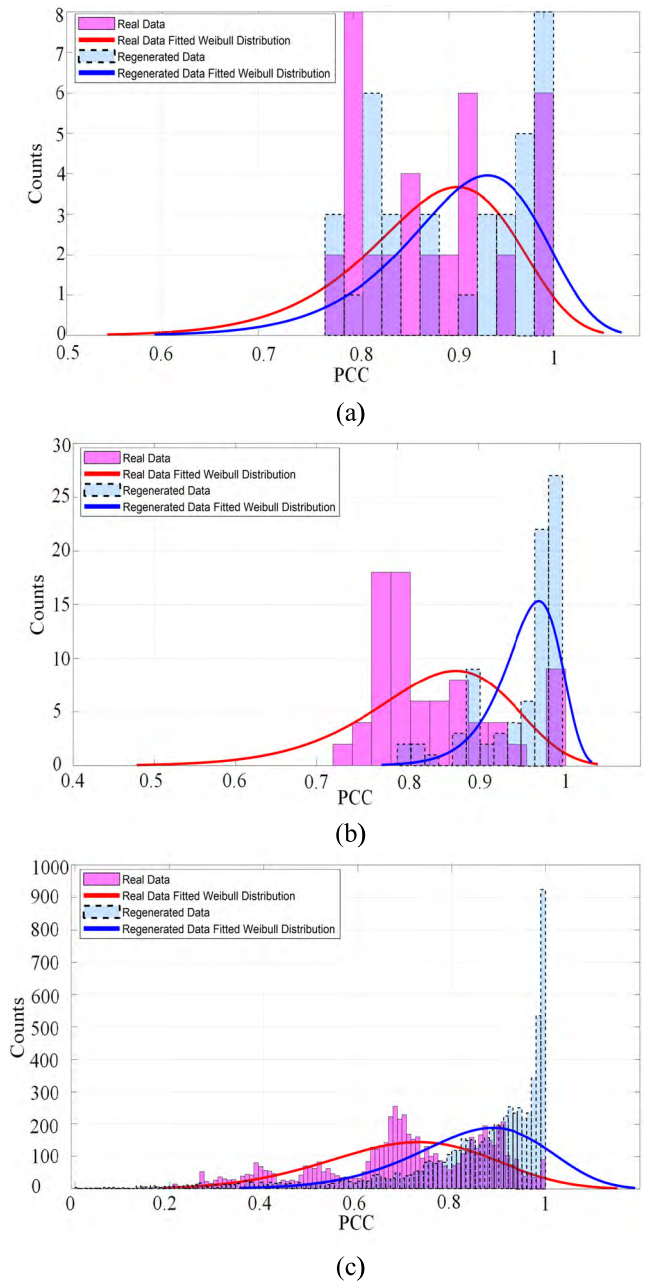


**FIGURE 12.** Pairwise PCC histograms of real and regenerated actions of CMU Mocap dataset a) jumping b) forward jump c) walking.

**TABLE 7.** Parameters of fitted distributions for real and regenerated actions of CMU Mocap dataset.

| Action | Real Data $k$ | Real Data $l$ | Regenerated Data $k$ | Regenerated Data $l$ | KL Divergence |
|---|---|---|---|---|---|
| Jumping | 0.9150 | 12.637 | 0.9456 | 14.0953 | 0.0843 |
| Forward jump | 0.8799 | 10.8301 | 0.9760 | 29.5768 | 2.2605 |
| Walking | 0.7684 | 4.6649 | 0.9073 | 6.9702 | 0.4577 |

Fig. 11b shows the PCC histogram of real and 90 regenerated movements for forward jump action of CMU dataset. Two-third of the PCC values are in the range of [0.930.99]. The pairwise PCC values of the real motions in Fig. 12b have greater standard deviation and lower values compared with PCC values of the regenerated data. The fitted shape and scale parameters, obviously show the situation. The results show that the MCDBN performs better in the jumping action of CMU dataset compared with forward jump action. The reason is the greater number of employed joints in the forward jump action and higher variation of joints positions especially in depth values of joints (the distance from the camera).

Most of the PCC values in the PCC histogram of real and 810 regenerated movements for walking action of CMU dataset (Fig. 12c) are in the range of [0.80.92]. The shapes of the PCC histograms of pairwise real and regenerated motions for walking action of CMU dataset are very similar and the parameters of fitted distributions are close to each other. Due to the complicatedness of walking action and other factors such as walking styles, the pairwise PCCs of real and regenerated motions for walking action have a very wide range. But, the mean and standard deviation of PCCs and the values of KL divergence of fitted Weibull distributions show the regenerated motions have good quality.

As can be seen in pairwise PCC histograms of real and regenerated motions for all actions of MHAD and CMU datasets, the regenerated motions have PCCs nearer to one. In other words, the real motions have more variations in comparison to regenerated motions and the regenerated motions seem artificial compared to real motions. This phenomenon is according to intuition because the real motions usually have some stochastic deviations due to humane factors.

Although there are some variations in the values of PCCs, the shapes of PCC histograms and the shapes of the pairwise PCCs histograms of real and regenerated motions, the PCCs ranges, their scatterings in different histograms bins and the statistics of distributions confirm that the MCDBN has learned to regenerate 3D motions for all the actions in both datasets accurately.

## VI. CONCLUSION AND FUTURE WORK

Regenerating human-like, realistic 3D human motions from 2D motions on the image plane is an ill-posed problem due to the ambiguities and number of possible projections. Considering coupled time series of 2D and 3D human motion as data modalities, in this paper, we propose a multimodal deep structure that is able to learn the cross-modal relationship of these modalities. The deep architecture named MCDBN consists of three distinct parts including two CDBNs for each data modalities and an MRBM for missing data generation. As the training procedure, the CDBNs extract compact representations from real-valued spatiotemporal patterns of 2D and 3D motion time series and the MRBM is trained based on the variation information criteria, in such a way that it would be able to regenerate compact representation of 3D motion data only given a compact representation of 2D motion data. Utilizing this property provides the ability of cross-modal data regeneration. While proposing a new evaluation criterion in the problem, the tests on two common datasets, Berkley

MHAD and CMU Mocap show that the model can regenerate 3D motions accurately and realistically and also the model achieved very outstanding quantitative results in terms of the proposed metric.

The results show that MCDBN achieves acceptable performance in 3D motion reconstruction. However, it is trained in an unsupervised manner and some useful kinematic and anthropometric constraints cannot be imposed directly. Finding methods which utilize powerful generative model and also have the capability of accepting some intuitive constraints can be the subjects of further studies. In addition, the hidden representation of the MRBM can be taught as the principal components of the input motion that will be useful in discriminative tasks such as action recognition. It seems that finding appropriate and justifiable evaluation metrics for the generative models in the motion modeling task and designing architecture that considers these metrics in the training and in the inference steps are the research problems that require more affords.

For the future work, we aim at applying the model in a more realistic situation and interactive environment like imitation learning or providing the ability to get raw videos as input and extracting required 2D and 3D motion data automatically which enable us to utilize the common monocular cameras as input sensor and the training data would not limit to special datasets.

## REFERENCES


[1] C.-H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5759–5767.

[2] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2D and 3D image cues for monocular body pose estimation," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2017, pp. 3941–3950.

[3] Y. Du *et al.*, "Marker-less 3D human motion capture with monocular image sequence and height-maps," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 20–36.

[4] X. Zhou, M. Zhu, S. Leonardos, and K. Daniilidis, "Sparse representation for 3D shape estimation: A convex relaxation approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1648–1661, Aug. 2017.

[5] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 186–201.

[6] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "MonoCap: Monocular human motion capture using a CNN coupled with a geometric prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 901–914, Apr. 2019.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[8] M. Meilă, "Comparing clusterings by the variation of information," in *Learning Theory and Kernel Machines*, vol. 3. Aug. 2003, pp. 173–187.

[9] S. Gold, C.-P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," in *Proc. Adv. Neural Inf. Process. Syst.*, 1995, pp. 957–964.

[10] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1263–1272.

[11] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of robot behavior using deep neural networks," *Robot. Automous Syst.*, vol. 62, no. 6, pp. 721–736, 2014.

[12] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox. (2018). "3D human pose estimation in RGBD images for robotic task learning." [Online]. Available: https://arxiv.org/abs/1803.02622

[13] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 689–696.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," *J. Mach. Learn. Res.*, vol. 15, pp. 2949–2980, Oct. 2014.

[16] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2222–2230.

[17] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Proc. Workshop Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Sep. 2014, pp. 474–490.

[18] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "ModDrop: Adaptive multi-modal gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1692–1706, Aug. 2016.

[19] K. Sohn, W. Shang, and H. Lee, "Improved multimodal deep learning with variation of information," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2141–2149.

[20] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "High-dimensional sequence transduction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 3178–3182.

[21] S. Sigtia, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, A. S. d'Avila Garcez, and S. Dixon, "A hybrid recurrent neural network for music transcription," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 2061–2065.

[22] R. Villegas, J. Yang, D. Ceylan, and H. Lee, "Neural kinematic networks for unsupervised motion retargetting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2018, pp. 8639–8648.

[23] D. Wu *et al.*, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, Aug. 2016.

[24] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine, "SFV: Reinforcement learning of physical skills from videos," in *Proc. SIGGRAPH Asia Tech. Papers*, Dec. 2018, p. 178.

[25] X. Lin and M. R. Amer. (2018). "Human motion modeling using DVGANs." [Online]. Available: https://arxiv.org/abs/1804.10652

[26] L. Bo and C. Sminchisescu, "Twin Gaussian processes for structured prediction," *Int. J. Comput. Vis.*, vol. 87, nos. 1–2, p. 28, 2010.

[27] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 283–298, Feb. 2008.

[28] M. R. Amer, T. Shields, B. Siddiquie, A. Tamrakar, A. Divakaran, and S. Chai, "Deep multimodal fusion: A hybrid approach," *Int. J. Comput. Vis.*, vol. 126, pp. 440–456, Apr. 2018.

[29] M. R. Amer, B. Siddiquie, S. Khan, A. Divakaran, and H. Sawhney, "Multimodal fusion using dynamic hybrid models," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2014, pp. 556–563.

[30] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[31] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognit. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.

[32] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," Dept. Comput. Sci., Univ. Colorado Boulder, Boulder, CO, USA, Tech. Rep., 1986.

[33] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," *Aistats*, vol. 10, pp. 33–40, Jan. 2005.

[34] T. Tieleman and G. Hinton, "Using fast weights to improve persistent contrastive divergence," in *Proc. 26th Annu. Int. Conf. Mach. Learn.* New York, NY, USA: ACM, Jun. 2009, pp. 1033–1040.

[35] D. P. Reichert, "Deep Boltzmann machines as hierarchical generative models of perceptual inference in the cortex," Tech. Rep., 2012.

[36] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[37] D. Wu, Y. Huang, H. Chen, Y. He, and S. Chen, "VPPAW penetration monitoring based on fusion of visual and acoustic signals using t-SNE and DBN model," *Mater. Des.*, vol. 123, pp. 1–14, Jun. 2017.

[38] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[39] I. Sutskever and G. Hinton, "Learning multilevel distributed representations for high-dimensional sequences," in *Artificial Intelligence and Statistics*, 2007, pp. 548–555.
[40] N. Garg and J. Henderson, "Temporal restricted Boltzmann machines for dependency parsing," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, Jun. 2011, pp. 11–17.
[41] G. W. Taylor, "Composable, distributed-state models for high-dimensional time series," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[42] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1345–1352.
[43] G. W. Taylor, L. Sigal, D. J. Fleet, and G. E. Hinton, "Dynamical binary latent variable models for 3d human pose tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 631–638.
[44] G. W. Taylor and G. E. Hinton, "Factored conditional restricted Boltzmann machines for modeling motion style," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 1025–1032.
[45] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two distributed-state models for generating high-dimensional time series," *J. Mach. Learn. Res.*, vol. 12, pp. 1025–1068, Mar. 2011.
[46] D. Hu, X. Li, and X. Lu, "Temporal multimodal learning in audiovisual speech recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3574–3582.
[47] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 899–907.
[48] Y. Bengio, E. Laufer, G. Alain, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2014, pp. 226–234.
[49] V. Mnih, H. Larochelle, and G. E. Hinton. (2012). "Conditional restricted boltzmann machines for structured output prediction." [Online]. Available: https://arxiv.org/abs/1202.3748
[50] I. Goodfellow, M. Mirza, A. Courville, and Y. Bengio, "Multi-prediction deep Boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 548–556.
[51] V. Upadhya and P. S. Sastry. (2017). "Learning RBM with a DC programming Approach." [Online]. Available: https://arxiv.org/abs/1709.07149
[52] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 53–60.
[53] *Carnegie Mellon University Motion Capture Database*. Accessed: Jan. 15, 2018. [Online]. Available: http://mocap.cs.cmu.edu/
[54] Accessed: Jan. 15, 2018. [Online]. Available: https://github.com/una-dinosauria/cmu-mocap
[55] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2878–2890, Dec. 2013.
[56] X. Chen and A. L. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1736–1744.
[57] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.
[58] A. Borji. (2018). "Pros and cons of GAN evaluation measures." [Online]. Available: https://arxiv.org/abs/1802.03446
[59] T. R. Derrick, B. T. Bates, and J. S. Dufek, "Evaluation of time-series data sets using the Pearson product-moment correlation coefficient," *Med. Sci. Sports Exerc.*, vol. 26, no. 7, pp. 919–928, 1994.
[60] A. K. Gayen, "The frequency distribution of the product-moment correlation coefficient in random samples of any size drawn from non-normal universes," *Biometrika*, vol. 38, nos. 1–2, pp. 219–247, 1951.
[61] C. Bauckhage. (2013). "Computing the Kullback-Leibler divergence between two Weibull distributions." [Online]. Available: https://arxiv.org/abs/1310.3713

**MUHAMMAD JAVAD HEYDARI** received the B.S. degree in computer engineering from Shahed University, in 2009, and the M.S. degree in computer engineering from the University of Tabriz, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Amirkabir University of Technology. His research interests include machine learning, deep learning, and machine vision.

**SAEED SHIRY GHIDARY** was born in Zanjan, Iran. He received the B.Sc. degree in electronic engineering and the M.Sc. degree in computer architecture from the Amirkabir University of Technology, in 1990 and 1994, respectively, and the Ph.D. degree in robotics and artificial intelligent systems from Kobe University, in 2002. He has been an Assistant Professor with the Amirkabir University of Technology, since 2004. His research interests include robotics, machine learning, machine vision, cognitive science, and brain modeling.